

Magic Square with Python

한신대학교 응용통계학과 201452024 박상희

1. 알고리즘

15	8	1	24	17	65
16	14	7	5	23	65
22	20	13	6	4	65
3	21	19	12	10	65
9	2	25	18	11	65
65	65	65	65	65	65

```
step 1 input n (an odd number), generate  $n \times n$  matrix M
step 2  $i = 0$ ;  $j = \text{int}((n+1)/2)-1$ ;  $k = 1$ ;  $M[i][j] = k$ 
step 3  $k = k + 1$ 
step 4  $i = i - 1$ ;  $j = j - 1$ 
step 5 check conditions
1.  $i < 0$  &  $j \geq 0 \Rightarrow i = n$ 
2.  $i \geq 0$  &  $j < 0 \Rightarrow j = n$ 
3.  $i < 0$  &  $j < 0 \Rightarrow i = i+2$  ;  $j = j + 1$ 
4.  $M[i][j] \neq 0 \Rightarrow i = i+2$  ;  $j = j + 1$ 
Step 6 repeat from step 3 until  $k = n*n$ 
```

2. Python Code

```
import numpy as np

def magic_square(n) :

    # 입력 받은 숫자가 짝수일 경우 에러 출력
    if n % 2 == 0 :
        print("Number is not odd!!!")
        return 0

    # nXn의 0으로 채워진 행렬 생성
    matrix = np.zeros((n,n))

    # 초기값 지정
    x, y, value = 0, int(n/2), 1

    # 시작 위치 1 지정
    matrix[x,y] = value

    # 나머지 칸 채우기 : 2 ~ 25
    for value in range(2, n*n+1) :

        # 위로 한 칸 상승, 왼쪽으로 한 칸 이동
        x, y = x-1, y-1

        # 더이상 왼쪽으로 이동할 수 없는 경우 반대편으로 이동
        if x < 0 and y >= 0 :
            x = n-1

        # 더 이상 위로 올라갈 수 없는 경우 반대편으로 이동
        elif x >= 0 and y < 0 :
            y = n-1

        # 가야할 자리가 맨 왼쪽 위 대각선인 경우 원래 자리에서 한 칸 아래로 이동
        elif x < 0 and y < 0 :
            x = x+2
            y = y+1

        # 가야할 곳이 빈자리가 아닌 경우 원래 자리에서 한 칸 아래로 이동
        elif matrix[x,y] != 0 :
            x = x+2
            y = y+1

        # 값 채워 넣기
        matrix[x,y] = value

    # 마방진 출력
    for i in range (n) :
        for j in range (n) :
            print ("%2d" %matrix[i][j], end="Wt")
        print ( )

    # 행렬 반환
    return matrix

# 5X5 마방진
magic_square(5)
```

3. 실행 결과

3X3 결과

6	1	8
7	5	3
2	9	4

5X5 결과

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11

7X7 결과

28	19	10	1	48	39	30
29	27	18	9	7	47	38
37	35	26	17	8	6	46
45	36	34	25	16	14	5
4	44	42	33	24	15	13
12	3	43	41	32	23	21
20	11	2	49	40	31	22